SH7080-Series E200F Emulator

Performance Measurement

Overview

The E200F emulator provides two functions to analyze the performance of user programs: on-chip performance analysis and AUD performance analysis. This document describes how to use the E200F emulator for the SH7080 in on-chip performance measurement.

The examples given in this document are of E200F-emulator usage in a stand-alone form. The functions described in this document are also available on all E200F emulators, regardless of the target device in use.

Contents

1. Specifications 2

2. Functional Descriptions 2

3. Software Preparation 2
3.1 Introduction 2
3.2 Installing the E200F Emulator Software 2
3.3 Auto-Update Utility 2

4. Operations 3
4.1 Activating the High-performance Embedded Workshop 3
4.2 Opening a Workspace 4
4.3 Analyzing Performance by Using Eventpoints 10
4.4 Analyzing Performance by Setting Data Conditions (Eventpoints) 23
4.5 Analyzing Performance by Measuring the Number of Executed Instructions 25

5. Frequently Asked Questions 27
5.1 Can you provide any notes on analyzing performance? 27

6. Related Documents 28

1. Specifications

The on-chip performance analysis function applies a counter in the device to measure the number of cycles from one specified condition being satisfied until another specified condition is subsequently satisfied. According to the supported device that is in use, various other items such as the number of cache misses and TLB misses may also be measurable. The on-chip performance analysis function cannot be used when the profiling function is enabled.

The AUD performance analysis function allows the user to measure times or numbers of instructions executed between specified events in the AUD event-detection system. The resolution of the timer can be set to 20 ns, 100 ns, 400 ns, or 1.6 µs. At 20 ns, the maximum time that can be measured is about six hours, and at 1.6 µs, the maximum time is about 20 days.

2. Functional Descriptions

This document explains how to analyze the performance by using the sample program included in the package that can be downloaded from the Renesas website.

The version of the tool is as follows:

Emulator Software:

E200F Emulator Debugger Version 2.04.00

3. Software Preparation

3.1 Introduction

Install the software provided in the CD-ROM of the E200F emulator to expand the sample program (tutorial workspace) to be used with this document on your personal computer.

The software in the CD-ROM of the E200F emulator can also be installed on a personal computer in which the High-performance Embedded Workshop has already been installed. In this case, some dialog boxes may be skipped in the installation process.

3.2 Installing the E200F Emulator Software

Execute HewInstMan.exe from the CD-ROM of the E200F emulator.

For details on installation, refer to the introductory guide for the E200F emulator on the Renesas website and follow the directions shown on the screen during installation. The full installation procedure is not described here.

3.3 Auto-Update Utility

When the auto-update utility is selected upon installation of software, it is possible to check the latest version of tools on the Internet.

4. Operations

This section explains how to activate the High-performance Embedded Workshop (HEW) and how to analyze the performance in the following steps.

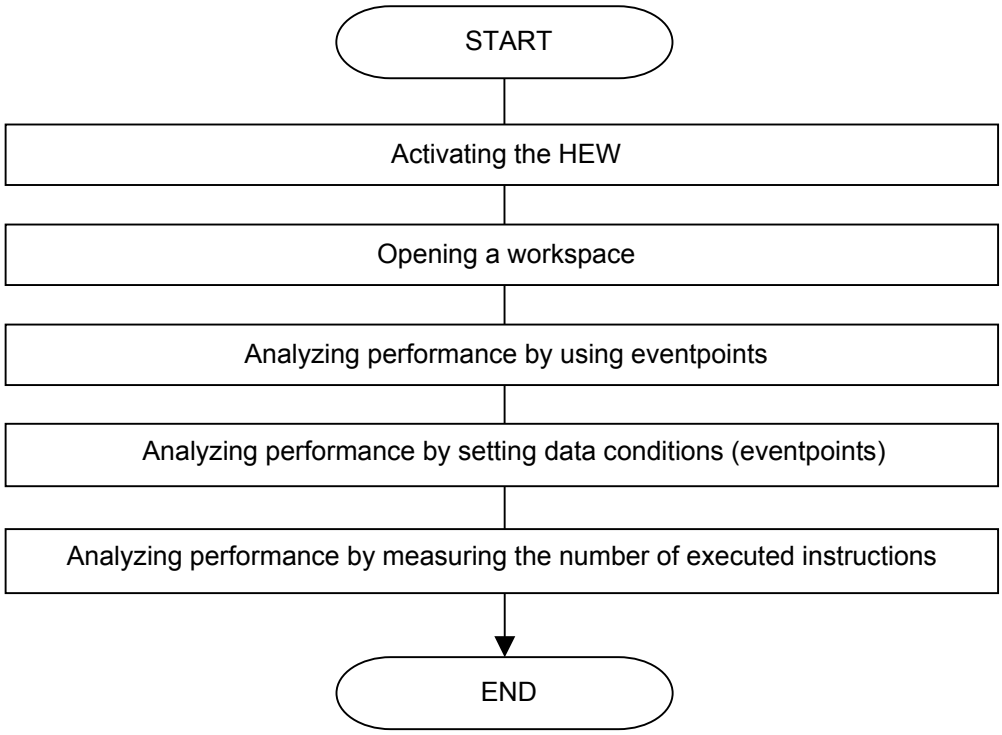
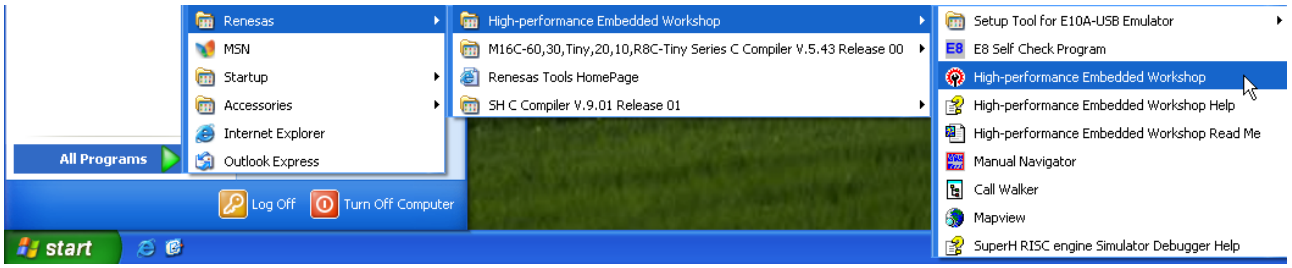

START
Activating the HEW
Opening a workspace
Analyzing performance by using eventpoints
Analyzing performance by setting data conditions (eventpoints)
Analyzing performance by measuring the number of executed instructions
END

Figure 4.1 Procedures for Sample Program Execution

4.1 Activating the High-performance Embedded Workshop

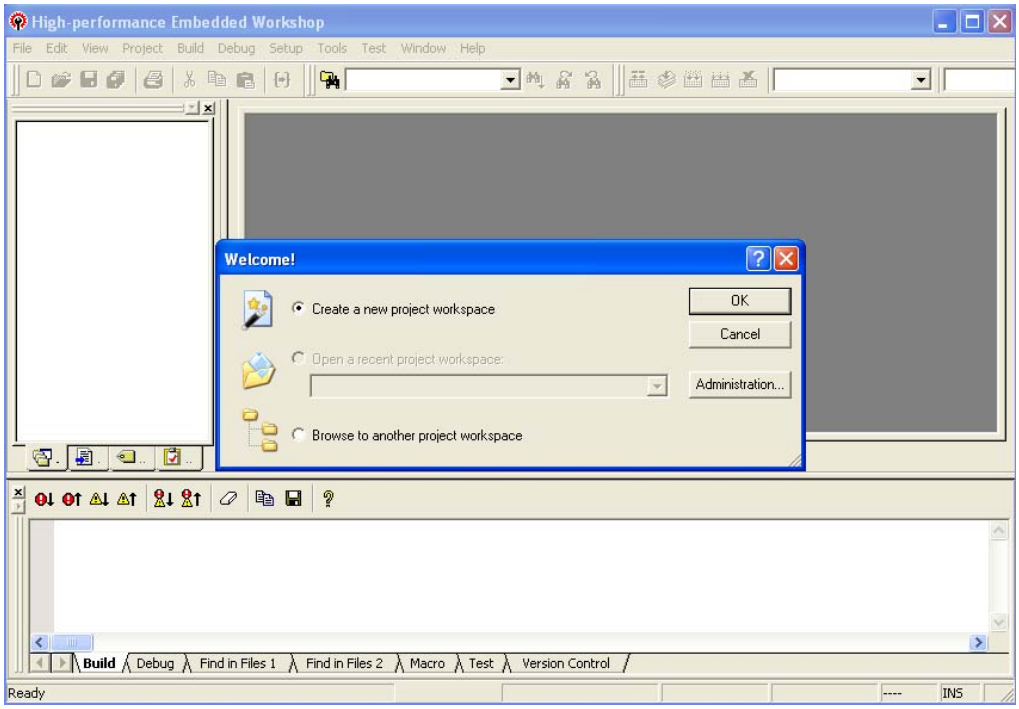
First, connect the E200F emulator with the user system to the host computer via the USB cable and check that debugging is enabled.

Next, activate the High-performance Embedded Workshop by opening the [Start] menu and selecting [All Programs], [Renesas], [High-performance Embedded Workshop], and [High-performance Embedded Workshop], in that order.

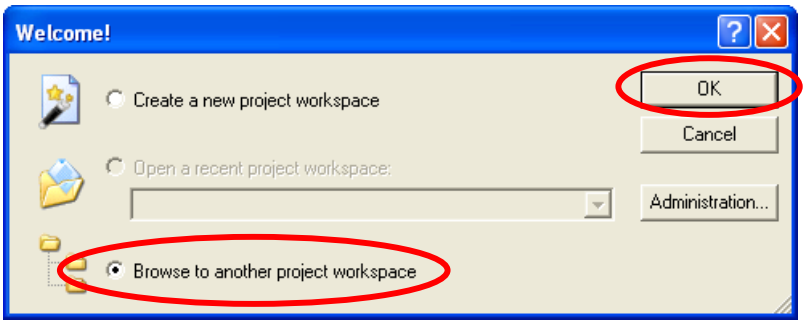


4.2 Opening a Workspace

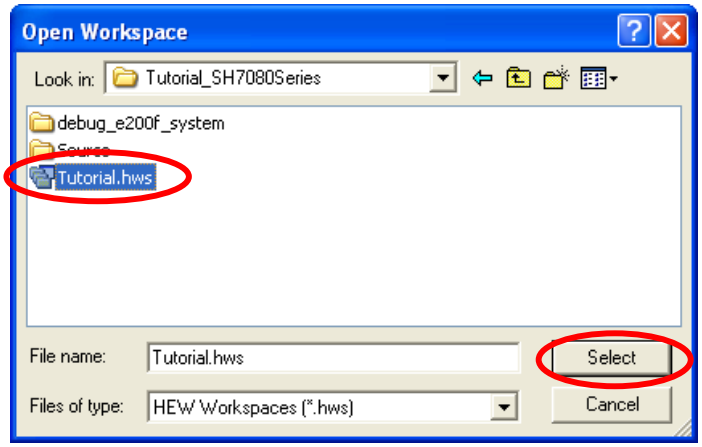
(1) The [Welcome!] dialog box will appear on the High-performance Embedded Workshop screen.



Select the [Browse to another project workspace] radio button in the [Welcome!] dialog box and click on the [OK] button.



(2) The [Open Workspace] dialog box will appear.



When the software from the CD-ROM of this product has been installed, workspace “Tutorial.hws” is stored in the folder structure shown below (standard location). Specify the correct location by opening the folders in order. Select the workspace “Tutorial.hws” and click on the [Open] button.

C:¥WorkSpace¥Tutorial¥E200F¥SH7080Series¥Tutorial_SH7080Series¥Tutorial.hws
C:¥WorkSpace └Tutorial └E200F └SH7080Series └Tutorial_SH7080Series └Tutorial.hws

Note: The above directory may not be specifiable depending on the software version. In this case, select the following directory.

<High-performance Embedded Workshop installation directory>

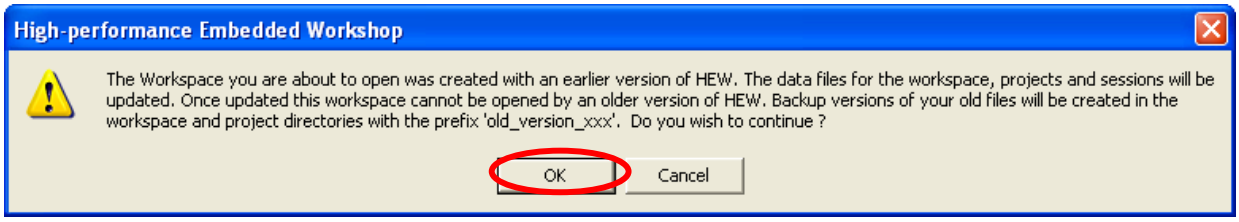
¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥SH7080Series¥Tutorial_SH7080Series

Directory examples:

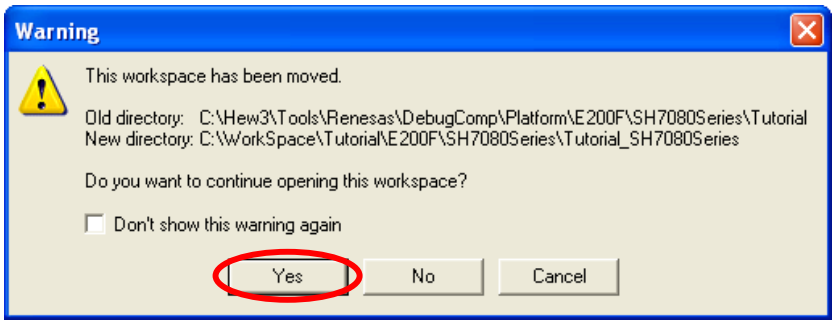
C:¥hew3¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥ SH7080Series¥Tutorial_SH7080Series

C:¥hew2¥Tools¥Renesas¥DebugComp¥Platform¥E200F¥ SH7080Series¥Tutorial_SH7080Series

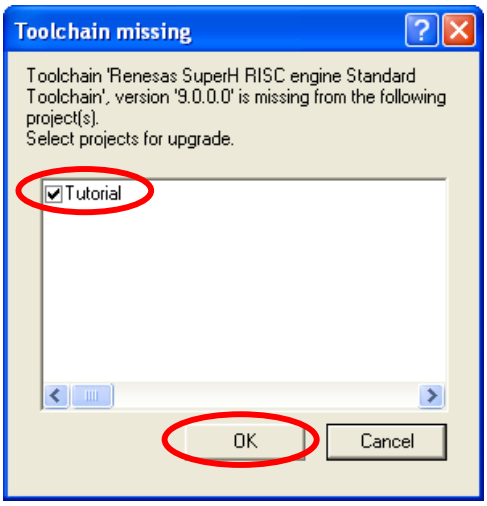
(3) f the workspace version is not the latest available, the following dialog box will appear. To update to the new version, click on the [OK] button.


High-performance Embedded Workshop
The Workspace you are about to open was created with an earlier version of HEW. The data files for the workspace, projects and sessions will be updated. Once updated this workspace cannot be opened by an older version of HEW. Backup versions of your old files will be created in the workspace and project directories with the prefix 'old_version_xxx'. Do you wish to continue ?
OK
Cancel

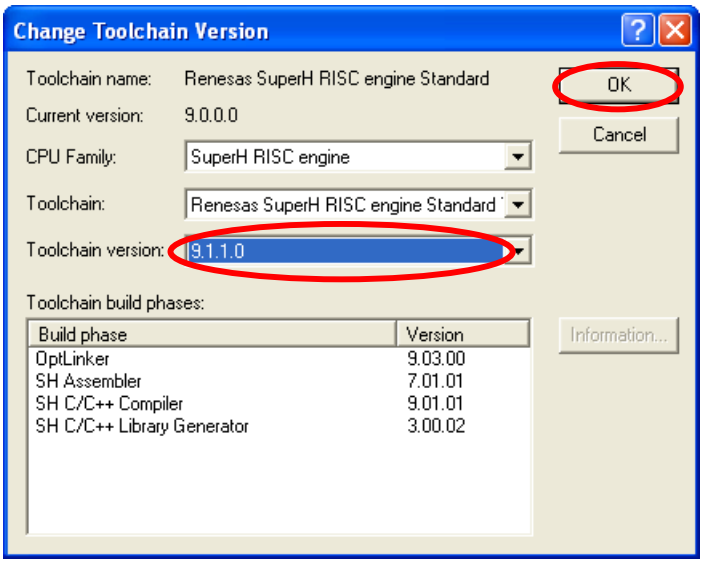
(4) A [Warning] dialog box appears. To open the workspace, which is now stored in a different directory, click on the [OK] button.



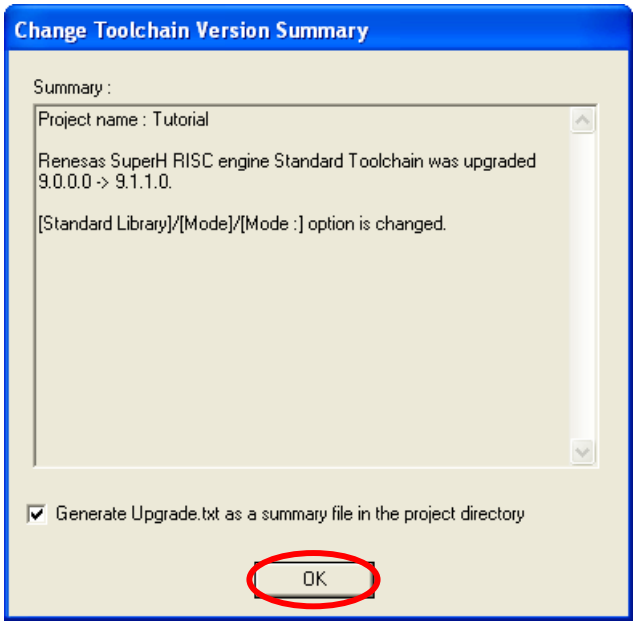
(5) If the [Toolchain missing] dialog box appears, select the target project name and click on the [OK] button.


Toolchain missing
Toolchain 'Renesas SuperH RISC engine Standard Toolchain', version '9.0.0.0' is missing from the following project(s).
Select projects for upgrade.
Tutorial
OK
Cancel

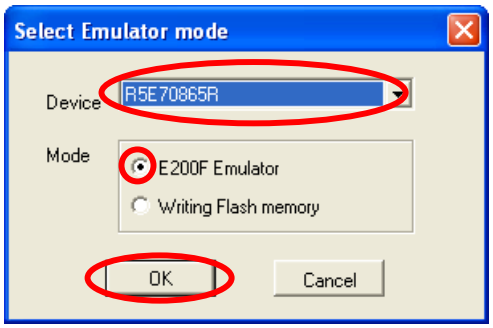
(6) If the [Changing Toolchain Version] dialog box appears, select the desired toolchain version and click on the [OK] button.



(7) If the [Change Toolchain Version Summary] dialog box appears, just click on the [OK] button.

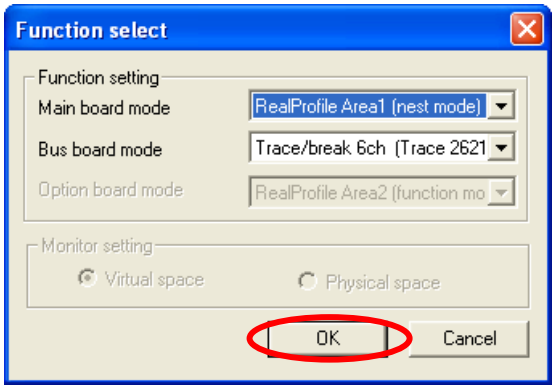

Change Toolchain Version Summary
Summary :
Project name : Tutorial
Renesas SuperH RISC engine Standard Toolchain was upgraded 9.0.0.0 -> 9.1.1.0.
[Standard Library]/[Mode]/[Mode :] option is changed.
Generate Upgrade.txt as a summary file in the project directory
OK

(8) Once the workspace is opened, the [Select Emulator mode] dialog box appears. Select the MCU in use from [Device]. Then select [E200 Emulator] for [Mode] and click on the [OK] button.

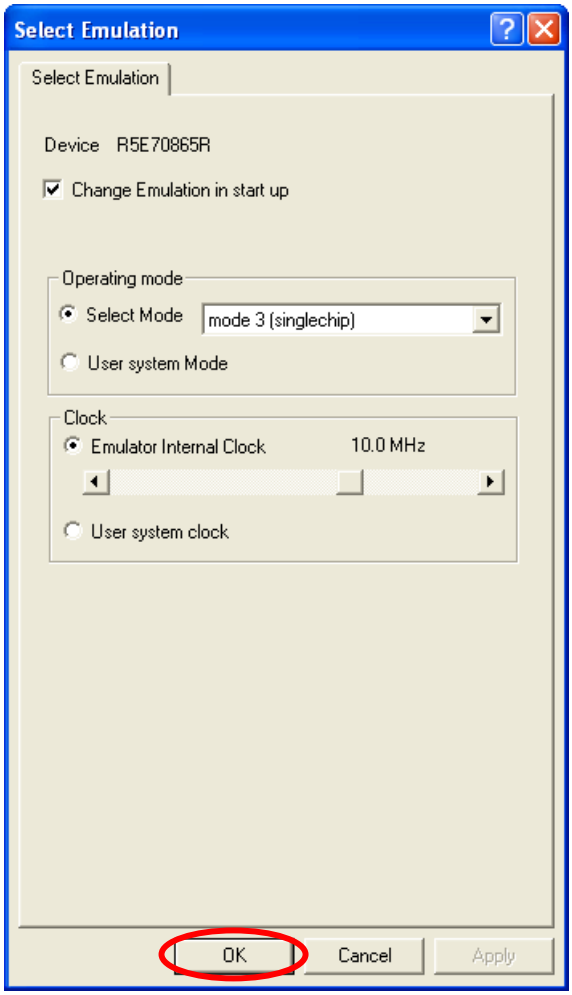


[R5E70865R] is selected in this example.

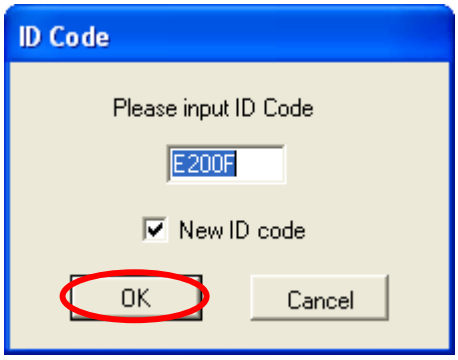
(9) The [Function select] dialog box appears. Click on the [OK] button to accept the default settings.



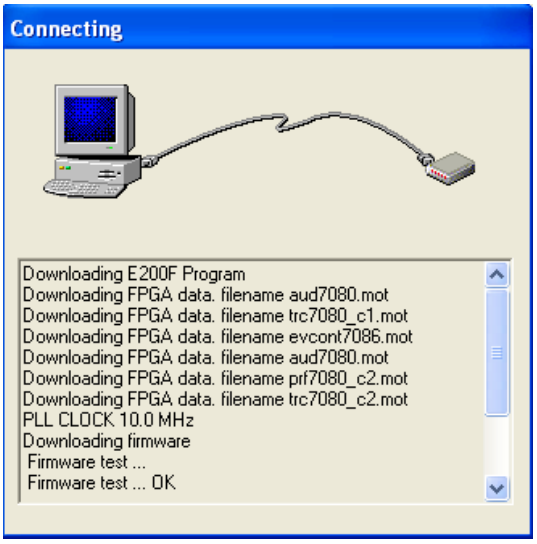
(10) The [Select Emulation] dialog box appears. Click on the [OK] button to accept the default settings.



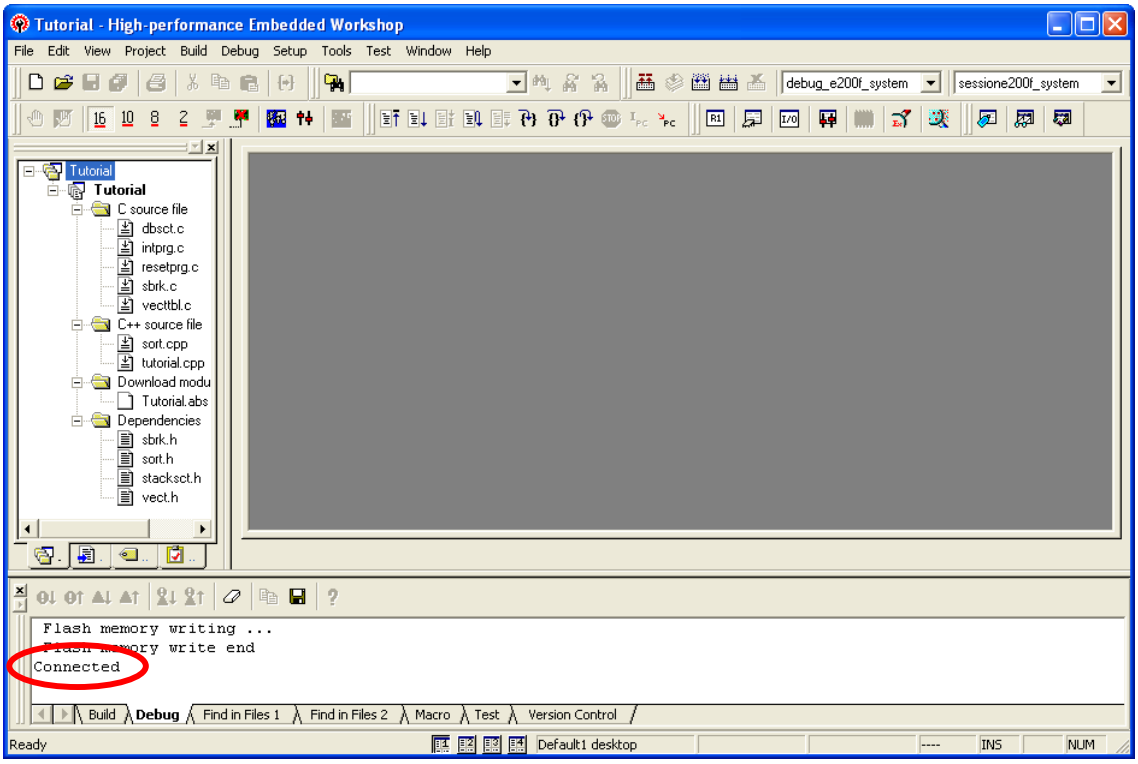
(11) The [ID Code] dialog box appears. Click on the [OK] button to accept the default setting, [E200F].



(12) While the E200F emulator is being connected, the [Connecting] dialog box shown below is displayed.



(13) After the connection has been established, operations on the High-performance Embedded Workshop screen become possible.

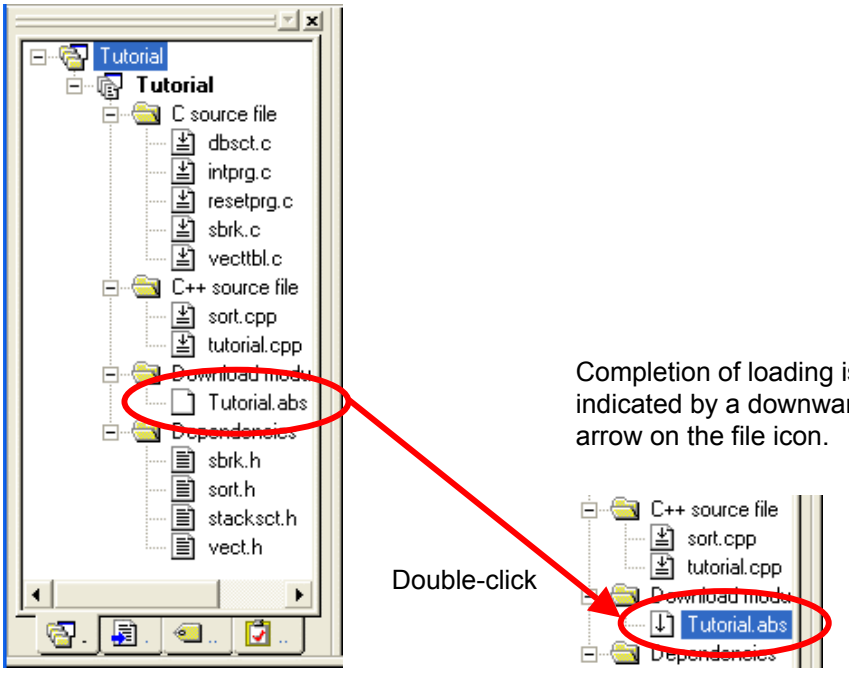


Once the E200F emulator is successfully connected, [Connected] is displayed on the [Debug] tabbed page of the [Output] pane.

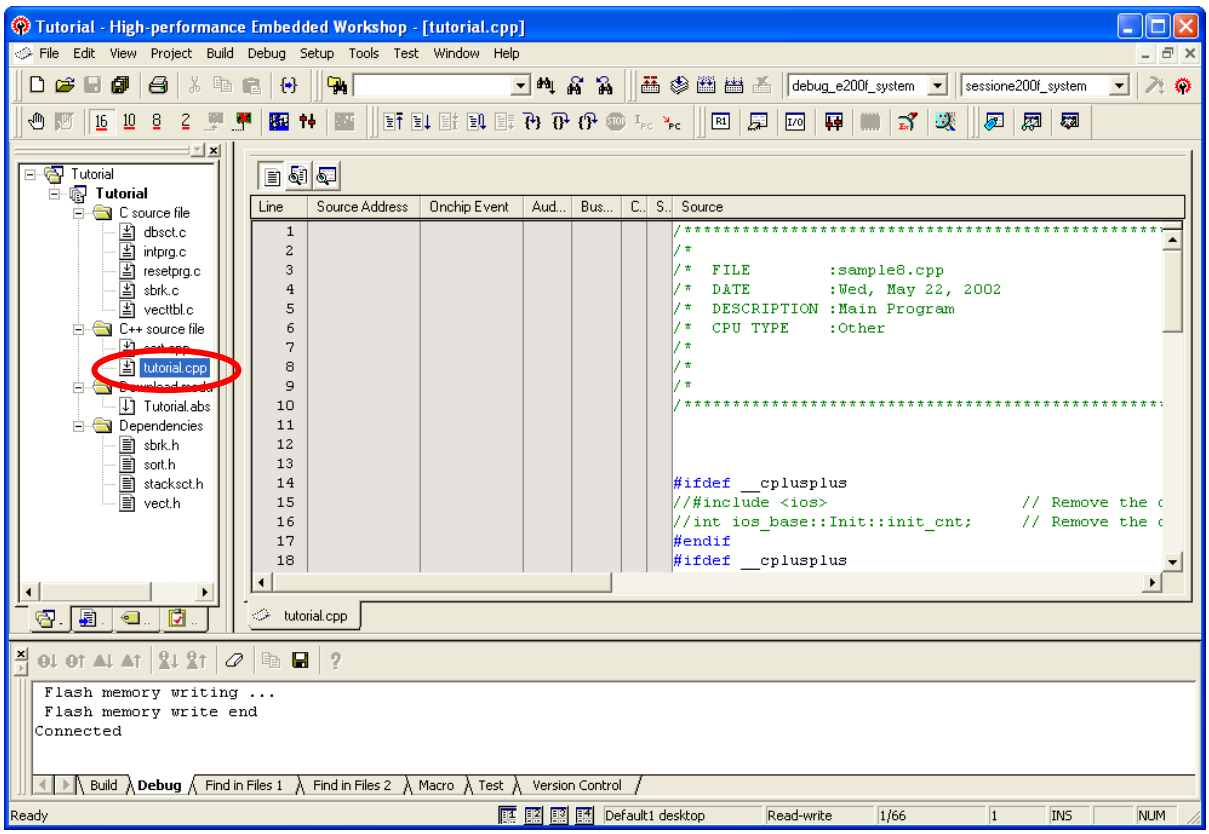
4.3 Analyzing Performance by Using Eventpoints

This section introduces two methods of performance analysis: measurement from the start of execution to a break and measurement over a specific period.

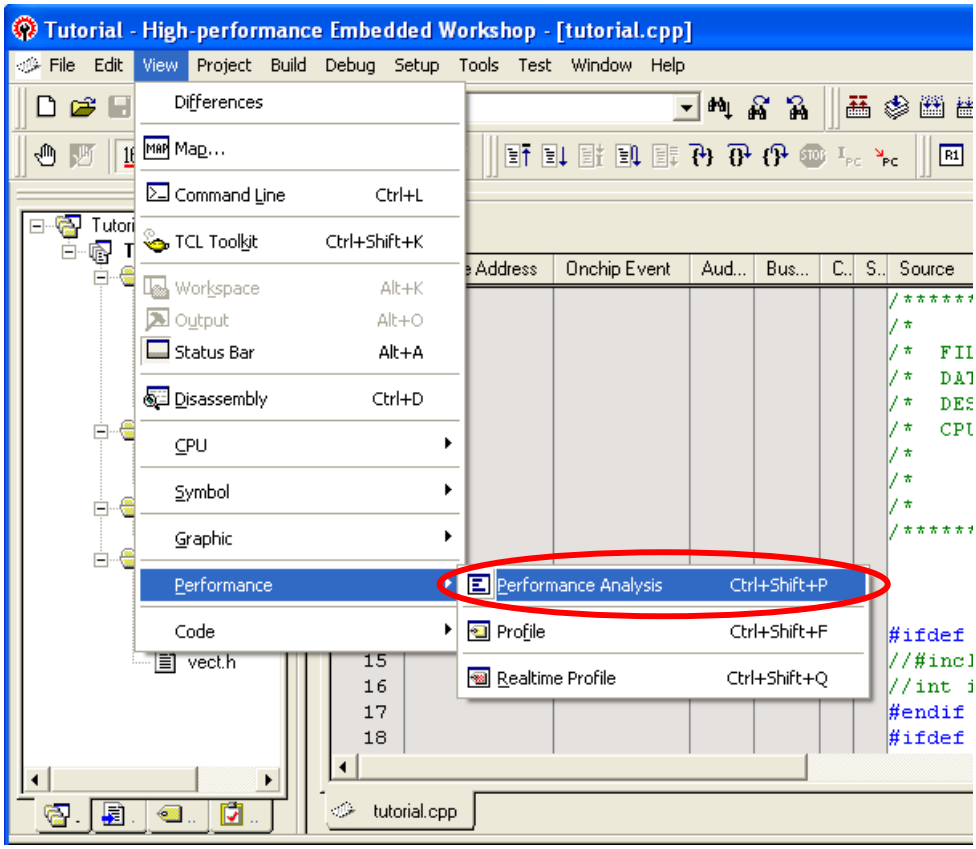
(1) To load a program, double-click on a load-module file (.abs) that has been registered in the workspace.



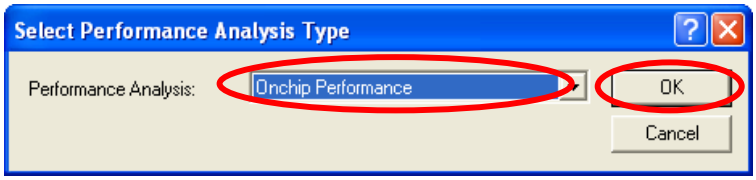
(2) Double-click on the source file name "tutorial.cpp" in the workspace to open the source code in the [Source] pane.


Tutorial - High-performance Embedded Workshop - [tutorial.cpp]
File Edit View Project Build Debug Setup Tools Test Window Help
Line Source Address Onchip Event Aud... Bus... C.. S.. Source
/*
/* FILE :sample8.cpp
/* DATE :Wed, May 22, 2002
/* DESCRIPTION :Main Program
/* CPU TYPE :Other
#ifdef __cplusplus
//#include <ios>
//int ios_base::Init::init_cnt;
#endif
#ifdef __cplusplus
tutorial.cpp
Flash memory writing ...
Flash memory write end
Connected
Build Debug Find in Files 1 Find in Files 2 Macro Test Version Control
Ready

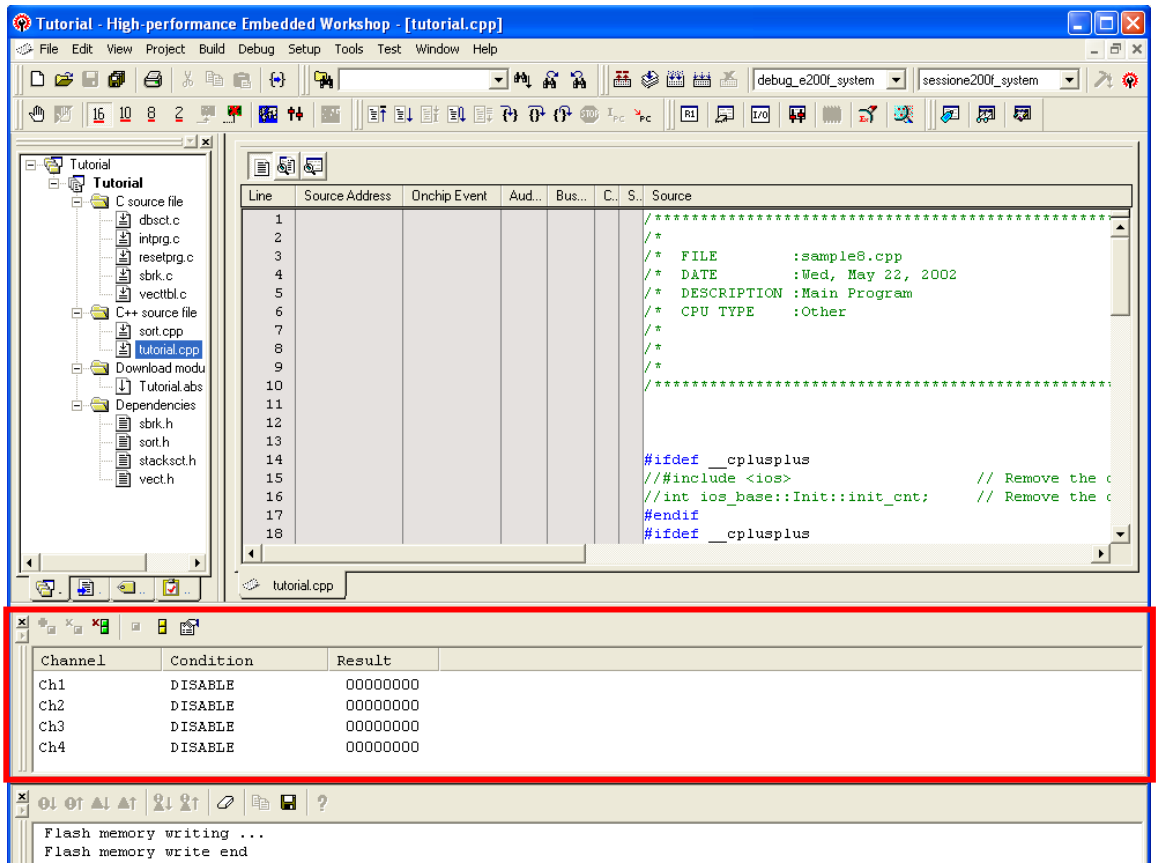
(3) Select [Performance Analysis] from the [Performance] item of the [View] menu.



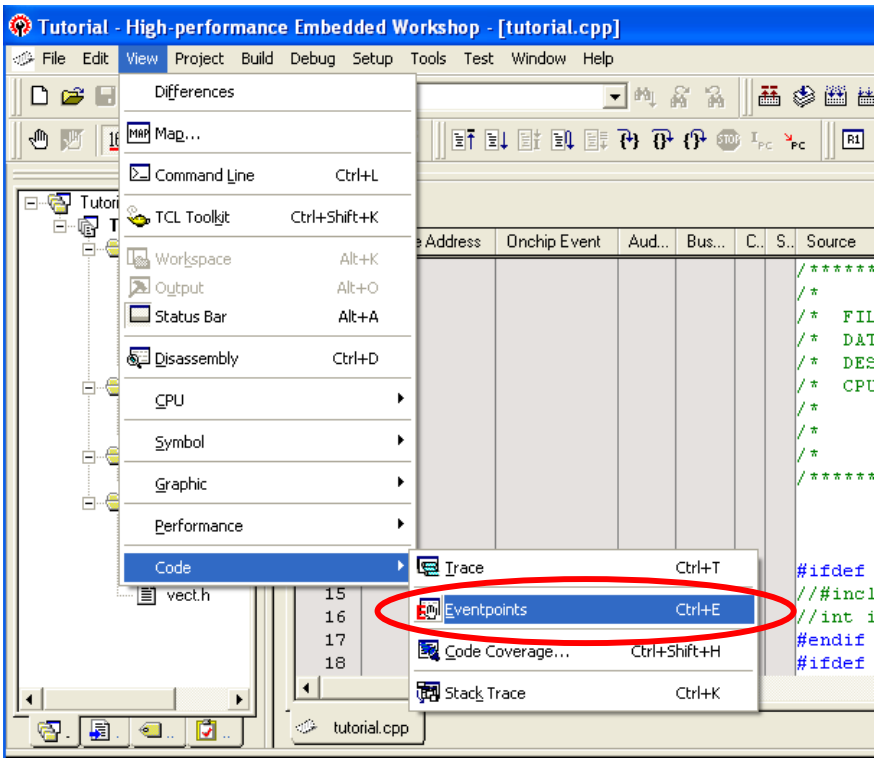
(4) The [Select Performance Analysis Type] dialog box will appear. Select [Onchip Performance] and click on [OK].



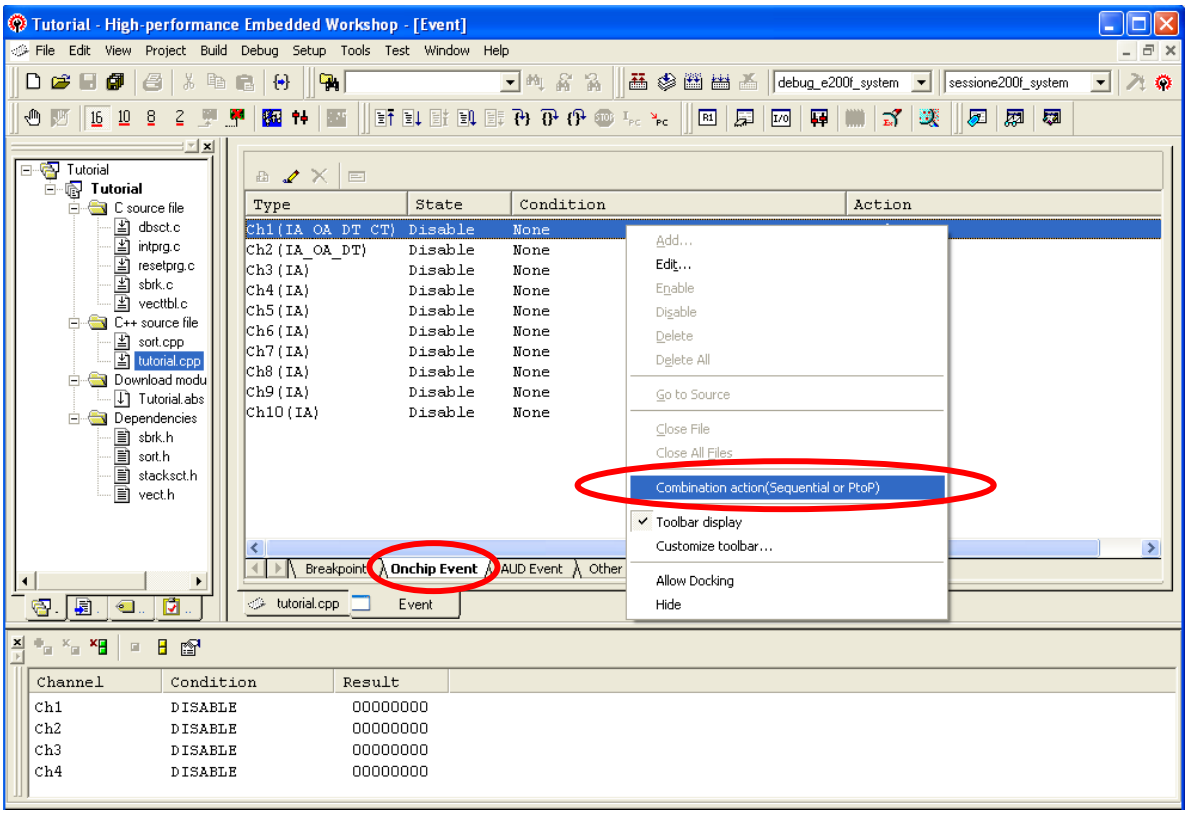
(5) The [Performance Analysis] pane will appear.



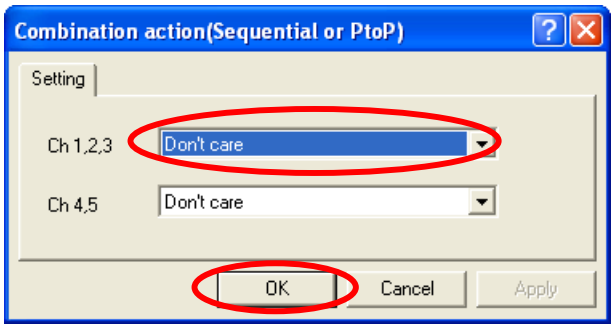
(6) Select [Eventpoints] from the [Code] item of the [View] menu.



(7) The [Event] pane is opened. Select the [Onchip Event] tab. Then right-click on a desired condition and select [Combination action(Sequential or PtoP)] from the popup menu.

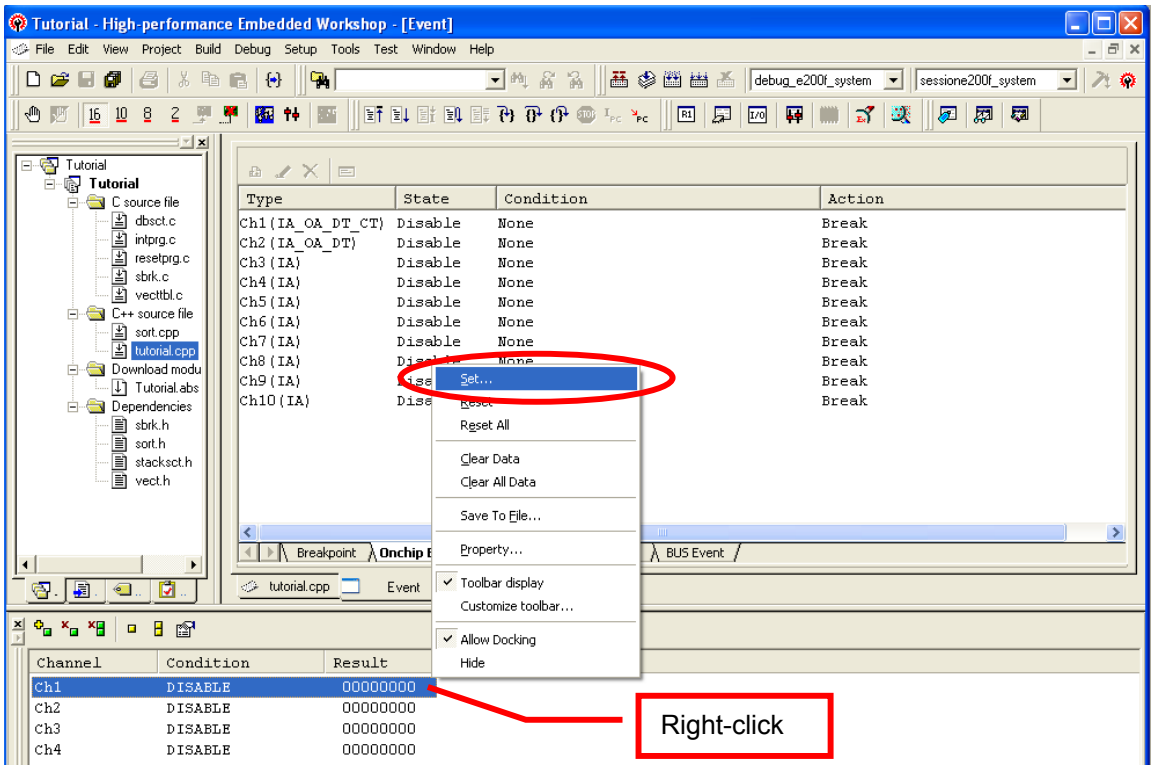


(8) The [Combination action(Sequential or PtoP)] dialog box will appear. Select [Don't care] for [Ch1,2,3] and click on [OK].

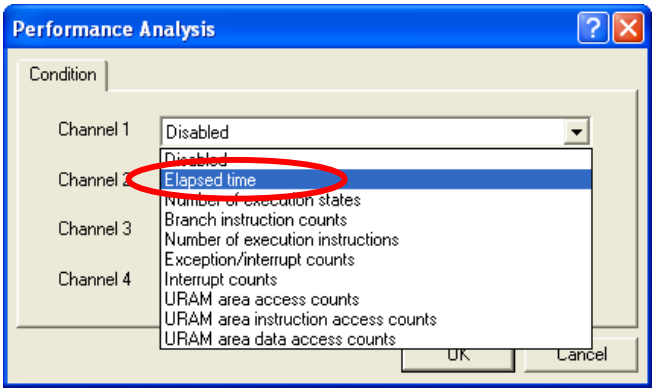


[Don't care]: Combinations of actions are not used as conditions.

(9) Right-click on the [Performance Analysis] pane and select [Set…] from the popup menu.



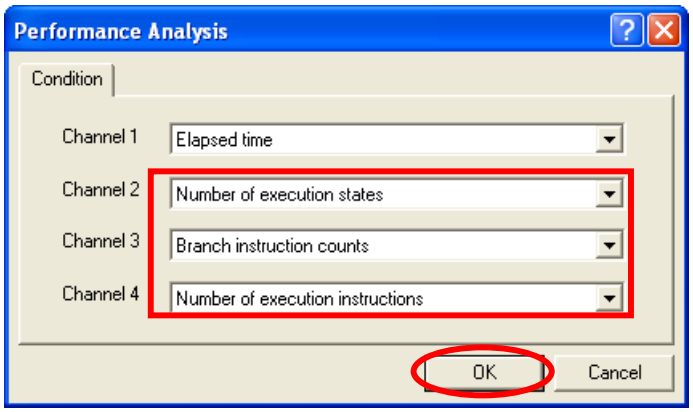
(10) The [Performance Analysis] dialog box opens. Select [Elapsed time] for [Channel 1].



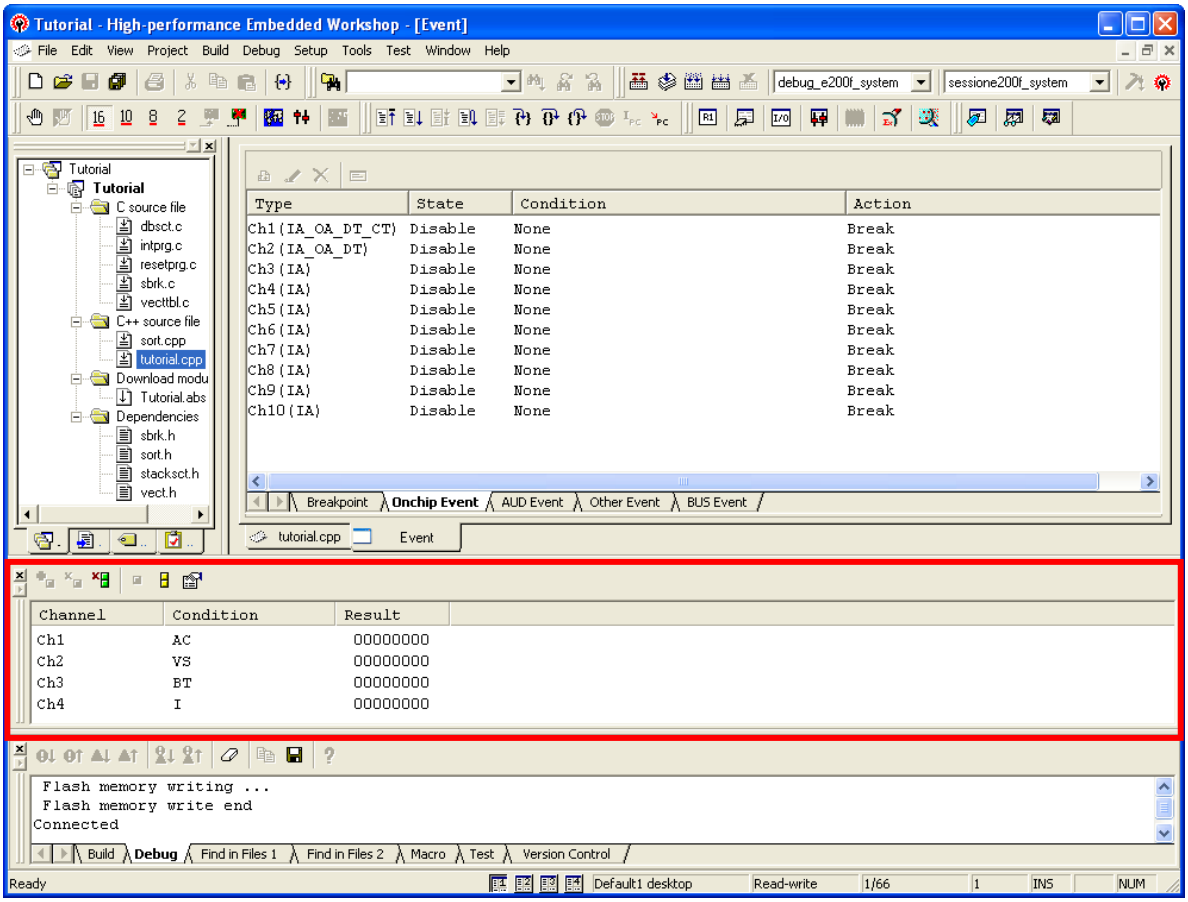
The following options are selectable in the [Performance Analysis] dialog box.

- Disabled (no option): Disables any setting
- Elapsed time (AC): Number of cycles (Iϕ) executed
- Number of execution states (VS): Number of execution states
- Branch instruction count (BT): Number of branch instructions
- Number of execution instructions (I): Number of executed instructions, including repeated execution
- Exception/interrupt counts (EA): Number of exceptions and interrupts
- Interrupt counts (INT): Number of interrupts
- URAM area access counts (UN): Numbers of times the URAM area was accessed for instructions and data
- URAM area instruction access counts (UIN): Number of times the URAM area was accessed for instructions
- URAM area data access counts (UDN): Number of times the URAM area was accessed for data

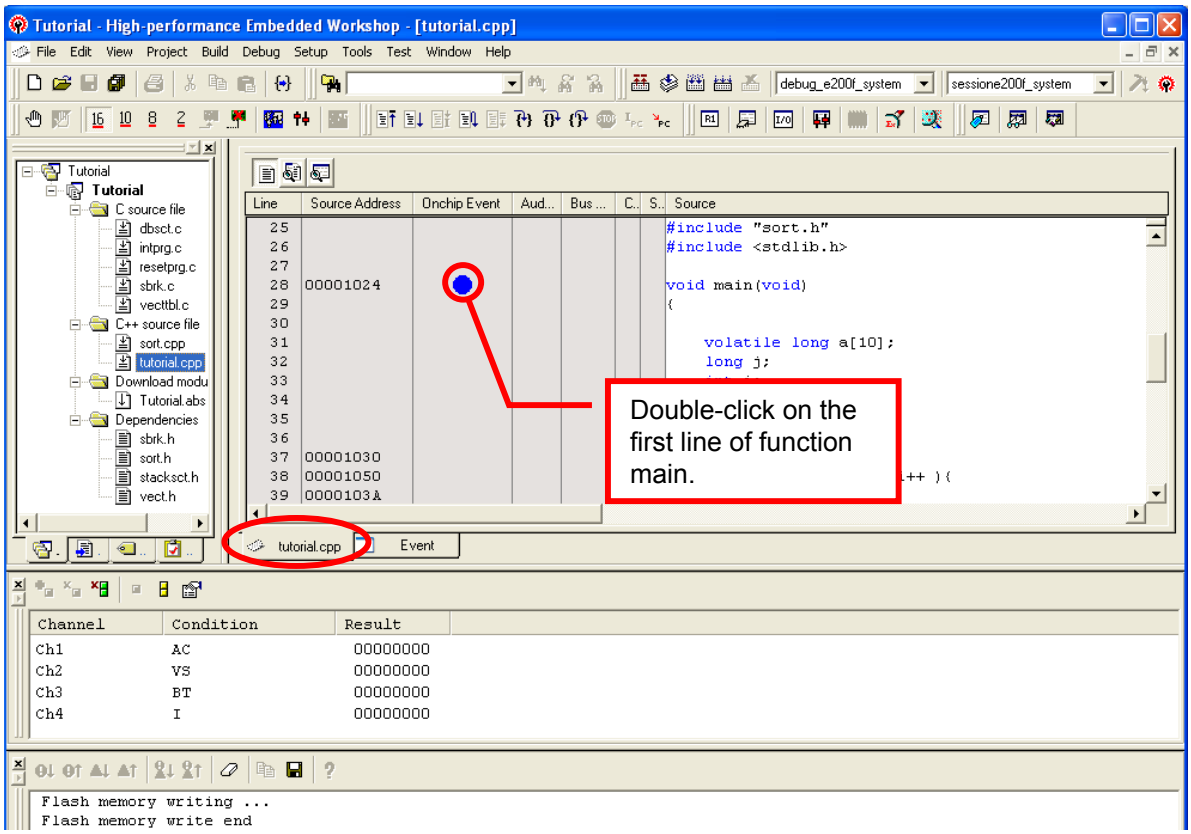
(11) Make settings for [Channel 2] to [Channel 4] as shown below and click on [OK].



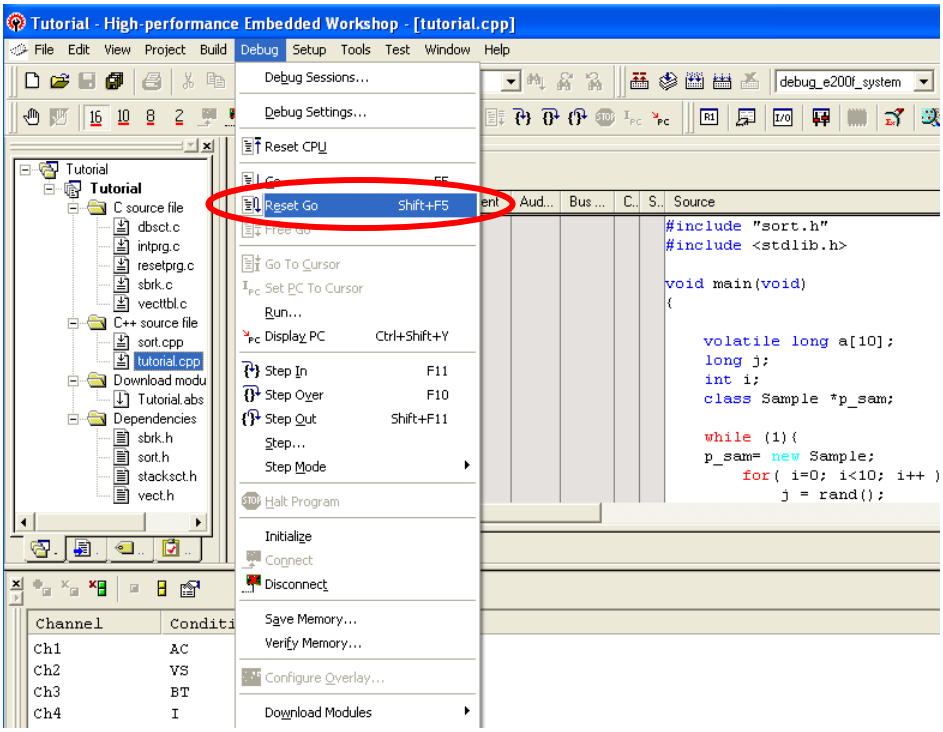
(12) The settings are shown in the [Performance Analysis] pane.


Tutorial - High-performance Embedded Workshop - [Event]
File Edit View Project Build Debug Setup Tools Test Window Help
Type State Condition Action
Ch1(IA_OA_DT_CT) Disable None Break
Ch2(IA_OA_DT) Disable None Break
Ch3(IA) Disable None Break
Ch4(IA) Disable None Break
Ch5(IA) Disable None Break
Ch6(IA) Disable None Break
Ch7(IA) Disable None Break
Ch8(IA) Disable None Break
Ch9(IA) Disable None Break
Ch10(IA) Disable None Break
Breakpoint Onchip Event AUD Event Other Event BUS Event
Channel Condition Result
Ch1 AC 00000000
Ch2 VS 00000000
Ch3 BT 00000000
Ch4 I 00000000
Flash memory writing ...
Flash memory write end
Connected
Build Debug Find in Files 1 Find in Files 2 Macro Test Version Control
Ready

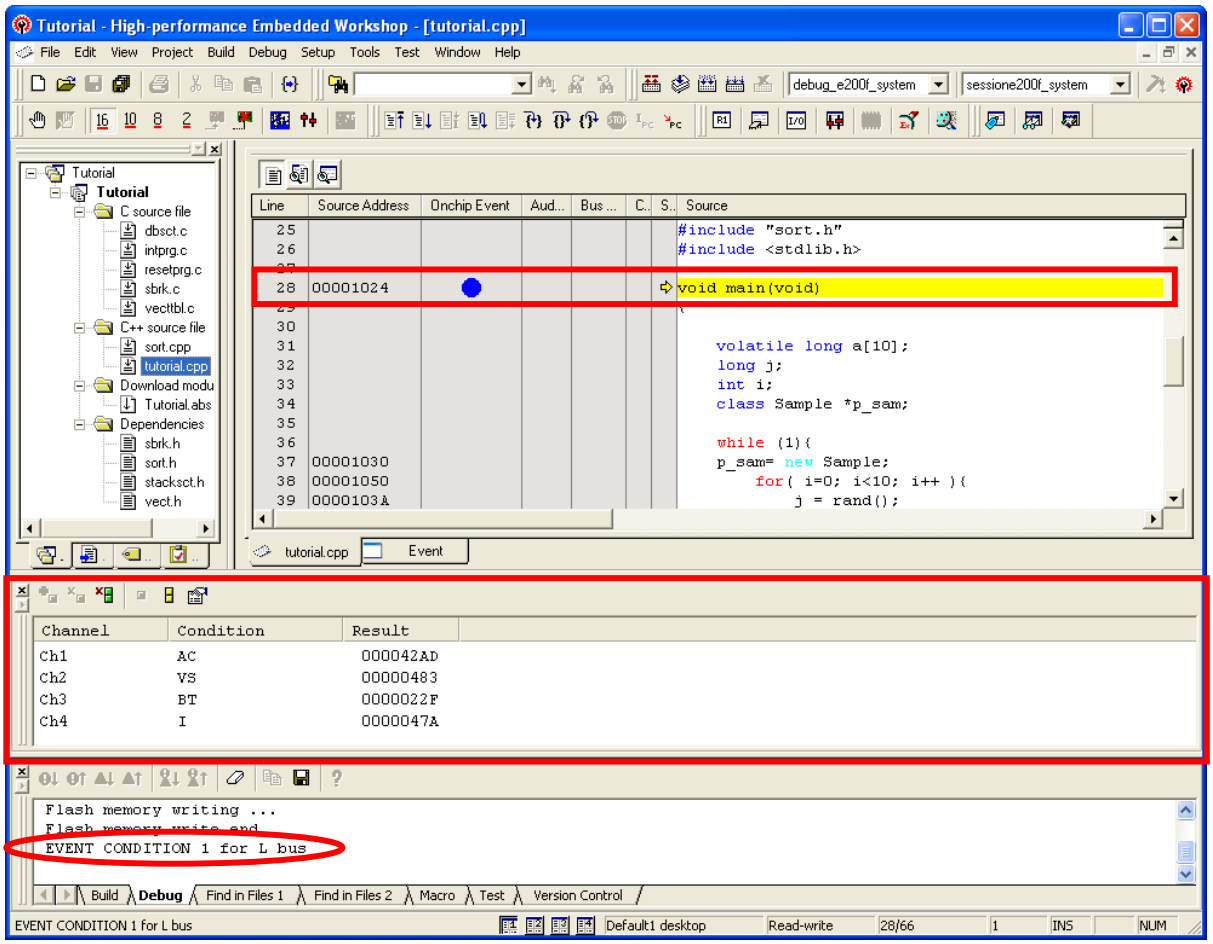
(13) Double-click on the first line of function main (line 28) in the source file “tutorial.cpp” to set an on-chip breakpoint.



(14) After setting the on-chip breakpoint, select [Reset Go] from the [Debug] menu to execute the program.



(15) The program stops at the line where the on-chip breakpoint has been set and the [Performance Analysis] pane shows the result of performance analysis. The comment “EVENT CONDITION for L bus” is displayed in the [Output] pane.



Now the performance from the reset routine to function main is displayed.

【Result of performance analysis】

- AC (number of cycles (Iϕ) executed): D’17069 (H’000042AD)

  Elapsed time = Number of cycles (Iϕ) executed x CPU clock period (Iϕ)

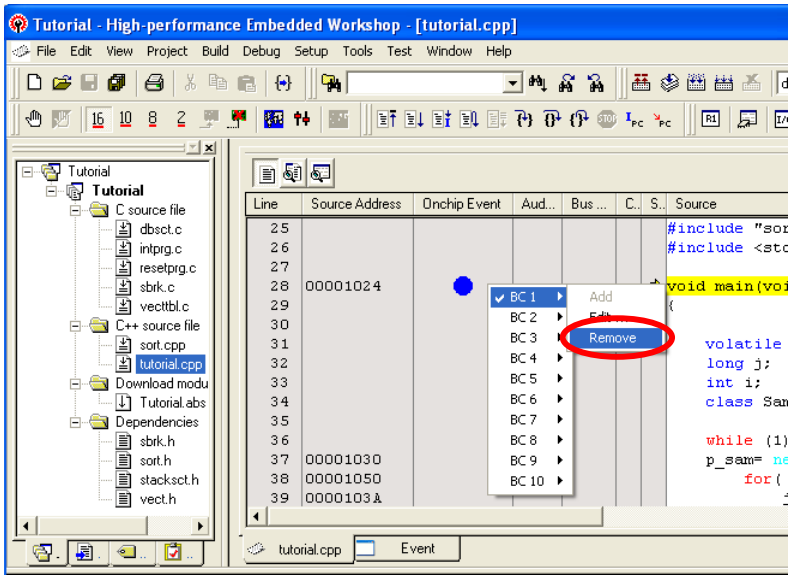
  = 17,069 x (1/10 MHz)

  = 1,707 μs

  （For details on setting of the CPU clock frequency, refer to step (10) of section 4.2.）

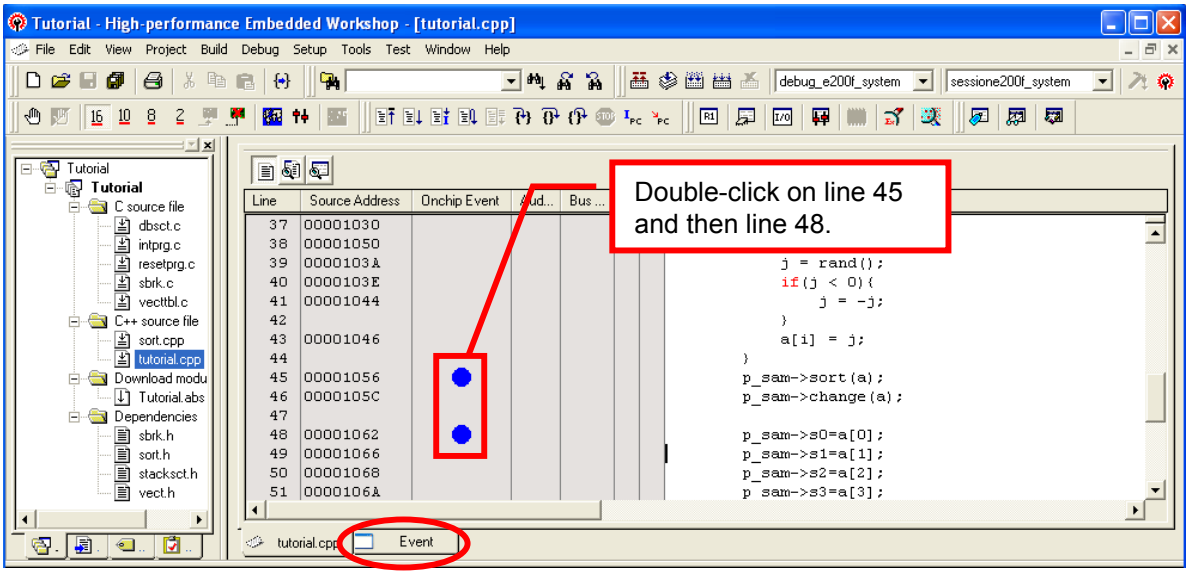
- VS (number of execution states): D’1155 (H’00000483)
- BT (number of branch instructions): D’559 (H’0000022F)
- I (number of executed instructions): D’1146 (H’0000047A)

(16) Delete condition BC1 from the [Onchip Event] column.

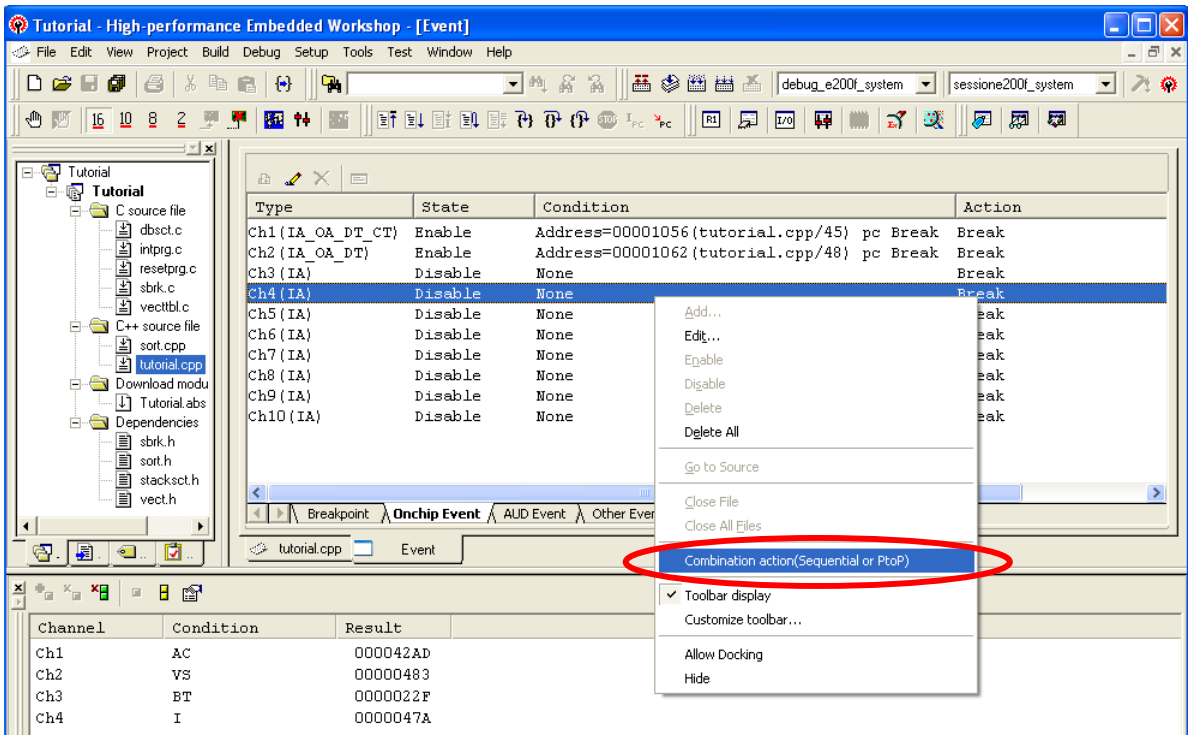


If BC2 has also been set, delete it too.

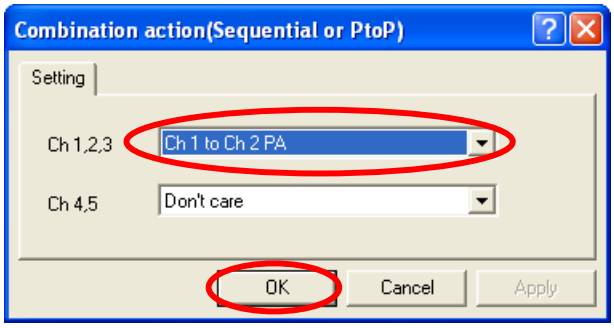
(17) Double-click on line 45 and then line 48. After that, click on the [Event] tab.



(18) Right-click on a desired condition in the [Event] pane to open the popup menu, and select [Combination action(Sequential or PtoP)].



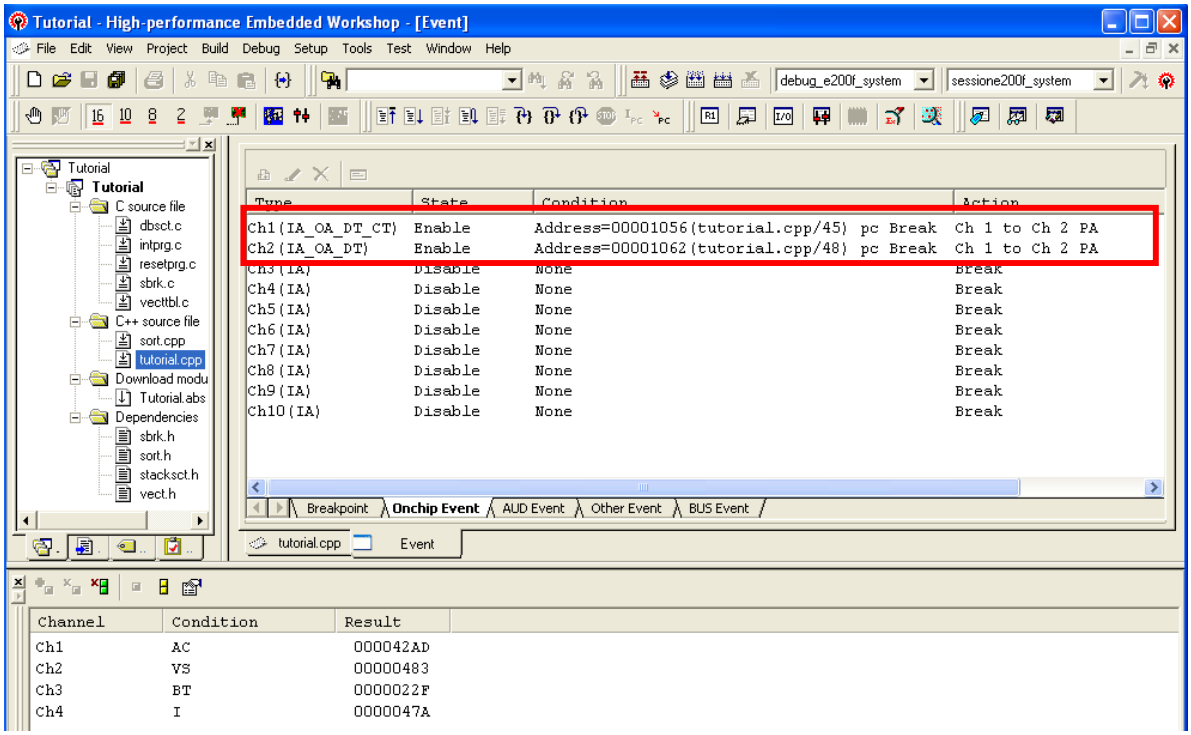
(19) The [Combination action(Sequential or PtoP)] dialog box will appear. Select [Ch 1 to Ch 2 PA] for [Ch1,2,3] and click on [OK].



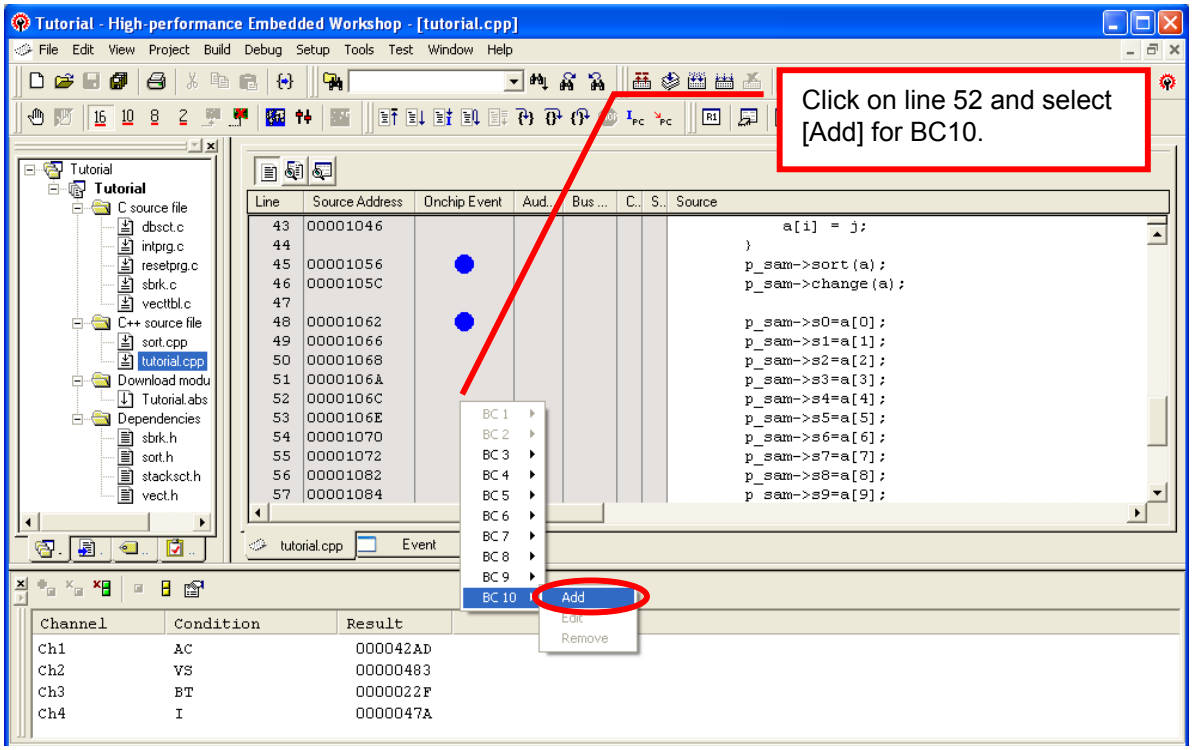
[Ch 1 to Ch 2 PA]: After event condition 1 (start condition) is satisfied, the emulator measures the performance until event condition 2 (end condition) is satisfied.

[Ch 2 to Ch 1 PA]: After event condition 2 (start condition) is satisfied, the emulator measures the performance until event condition 1 (end condition) is satisfied.

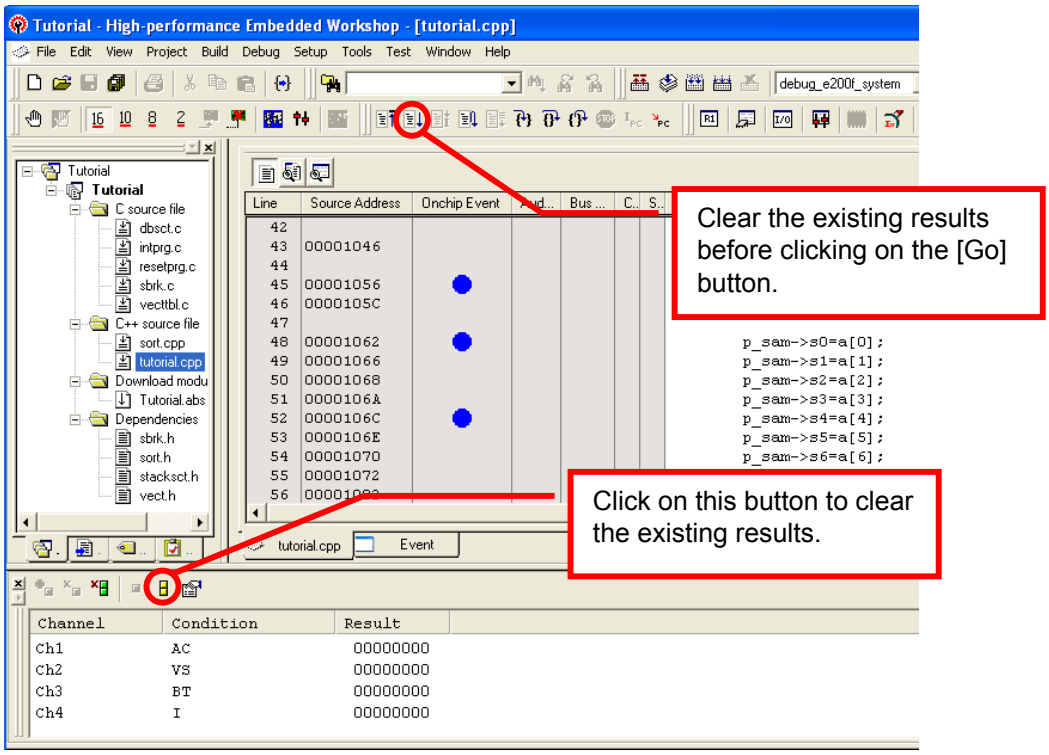
(20) The [Event] pane shows the settings that have been made.


Tutorial - High-performance Embedded Workshop - [Event]
Type State Condition Action
Ch1(IA_OA_DT_CT) Enable Address=00001056(tutorial.cpp/45) pc Break Ch 1 to Ch 2 PA
Ch2(IA_OA_DT) Enable Address=00001062(tutorial.cpp/48) pc Break Ch 1 to Ch 2 PA
Ch4(IA) Disable None Break
Ch5(IA) Disable None Break
Ch6(IA) Disable None Break
Ch7(IA) Disable None Break
Ch8(IA) Disable None Break
Ch9(IA) Disable None Break
Ch10(IA) Disable None Break
Breakpoint Onchip Event AUD Event Other Event BUS Event
Channel Condition Result
Ch1 AC 000042AD
Ch2 VS 00000483
Ch3 BT 0000022F
Ch4 I 0000047A

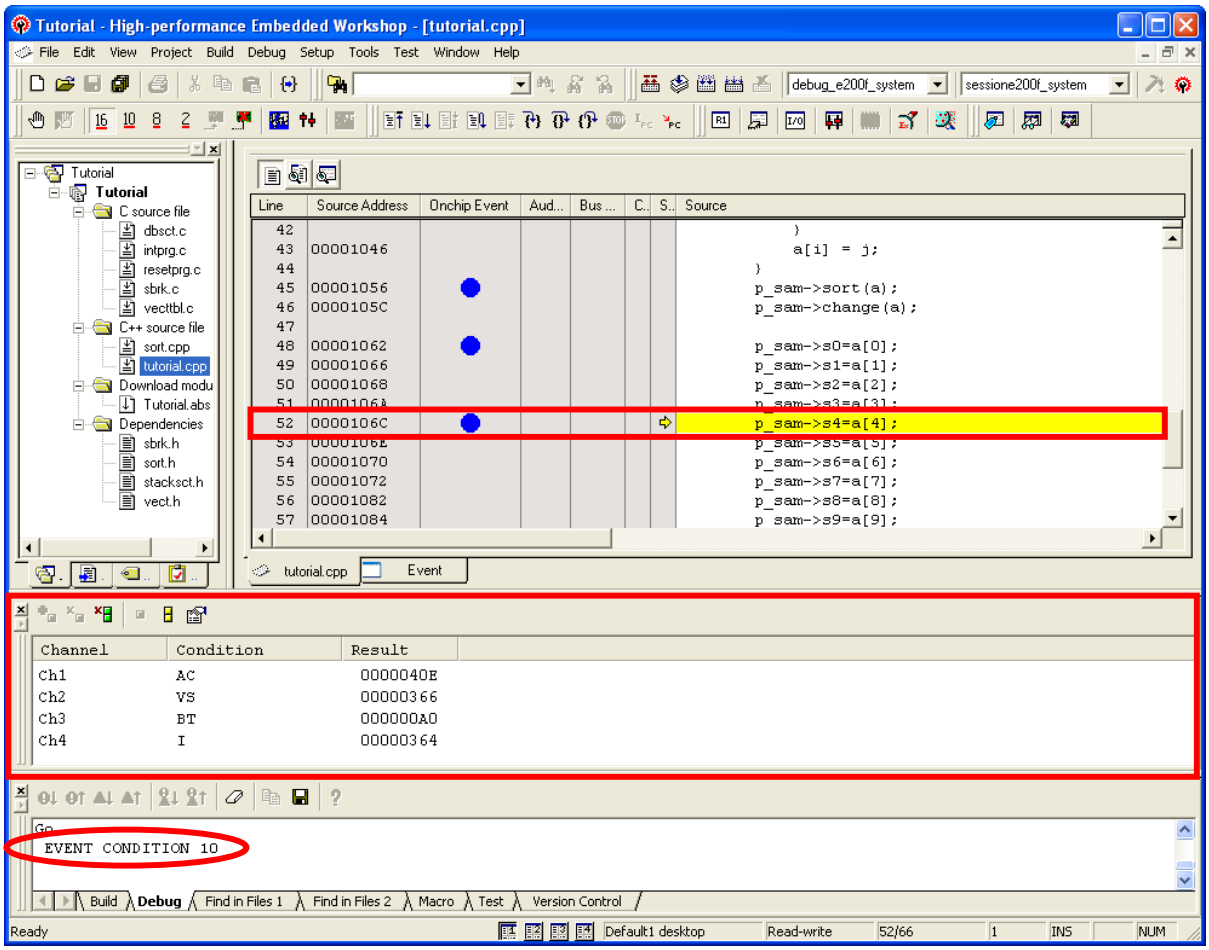
(21) Select the [tutorial.cpp] tab. Right-click on line 52 and select [Add] for BC10.



(22) Clear the result of performance analysis in the [Performance Analysis] pane and then execute the program.



(23) The program stops at line 52, where the breakpoint has been set, and the [Performance Analysis] pane shows the results of performance analysis. The [Output] pane shows the comment “EVENT CONDITION 10”.

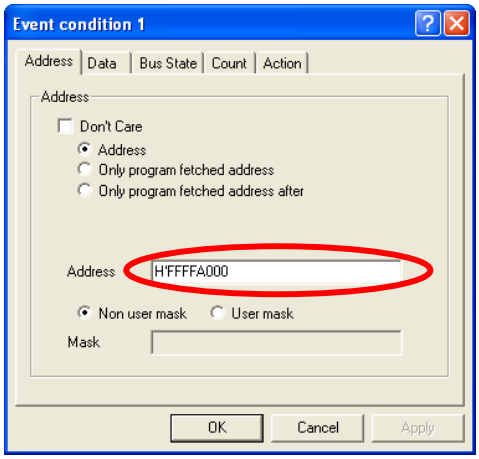


The performance from line 45 to 48, from which the functions sort and change are called, in that order, has been analyzed.

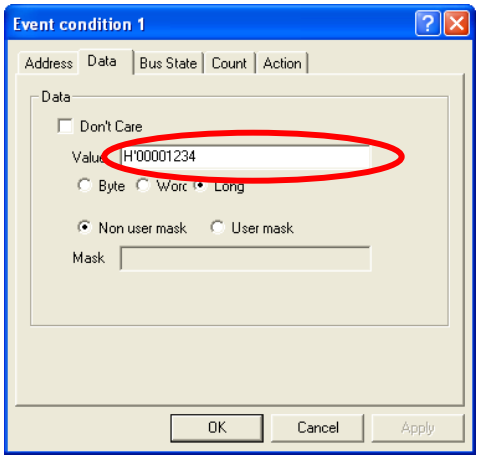
4.4 Analyzing Performance by Setting Data Conditions (Eventpoints)

This section gives examples of setting data conditions.

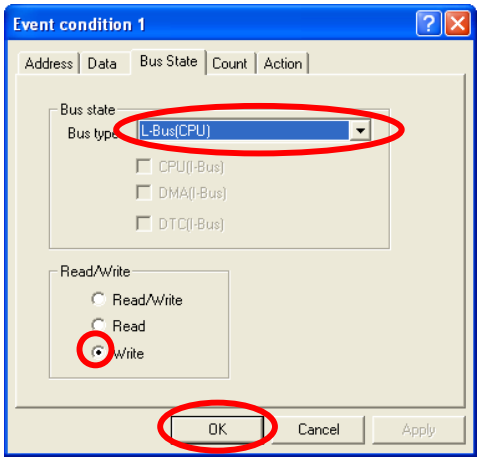
(1) In this example, the condition is the writing of a particular value (32 bits: H'00001234) into a variable (at FFFFA000 in RAM). Double-clicking on the [Ch1] column in the [Event] pane opens the [Event condition 1] dialog box. Click on the [Address] tab and specify H'FFFFA000 as [Address].



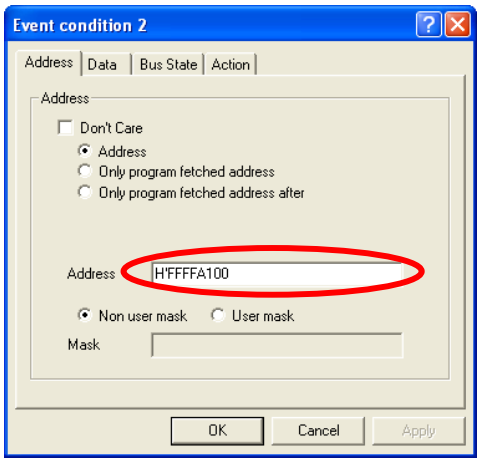
Click on the [Data] tab. Remove the tick from the [Don't Care] checkbox and specify H'00001234 as [Value].



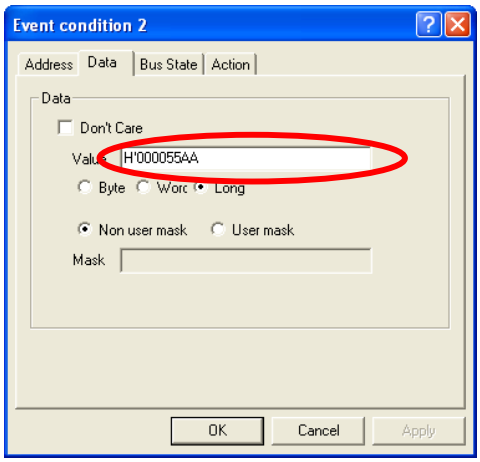
Click on the [Bus State] tab. Select [L-Bus(CPU)] as [Bus type] and [Write] for [Read/Write], and then click on [OK].



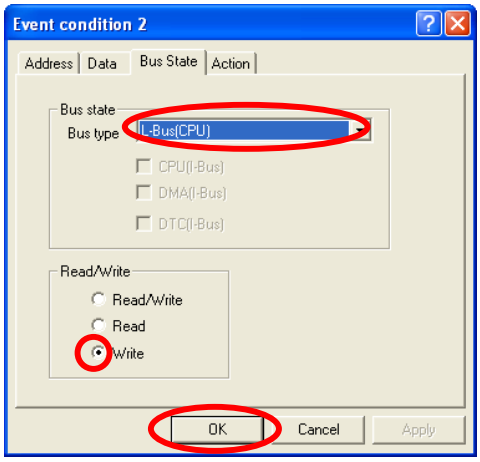
(2) In the next example, the condition is the writing of a particular value (H'000055AA) into another variable (at FFFFA100 in RAM). Double-clicking on the [Ch2] column in the [Event] pane opens the [Event condition 2] dialog box. Click on the [Address] tab and specify H'FFFFA100 as [Address].



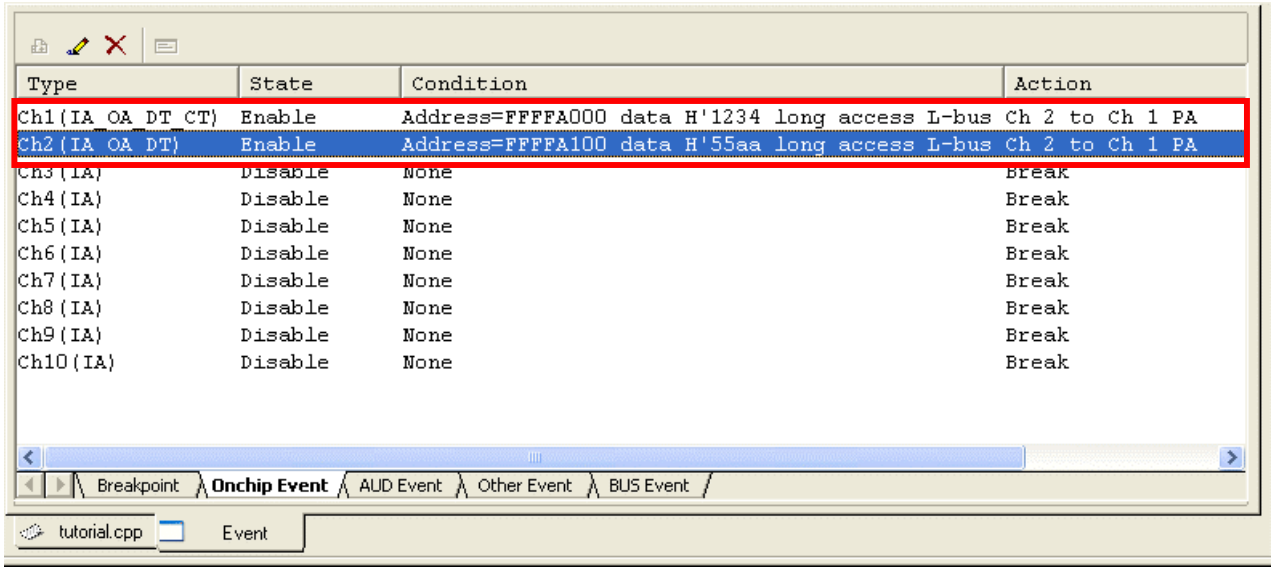
Click on the [Data] tab. Specify H'000055AA as [Value].



Click on the [Bus State] tab. Select [L-Bus(CPU)] as [Bus type] and [Write] for [Read/Write], and then click on [OK].



(3) The [Performance Analysis] pane shows the settings that have been made.

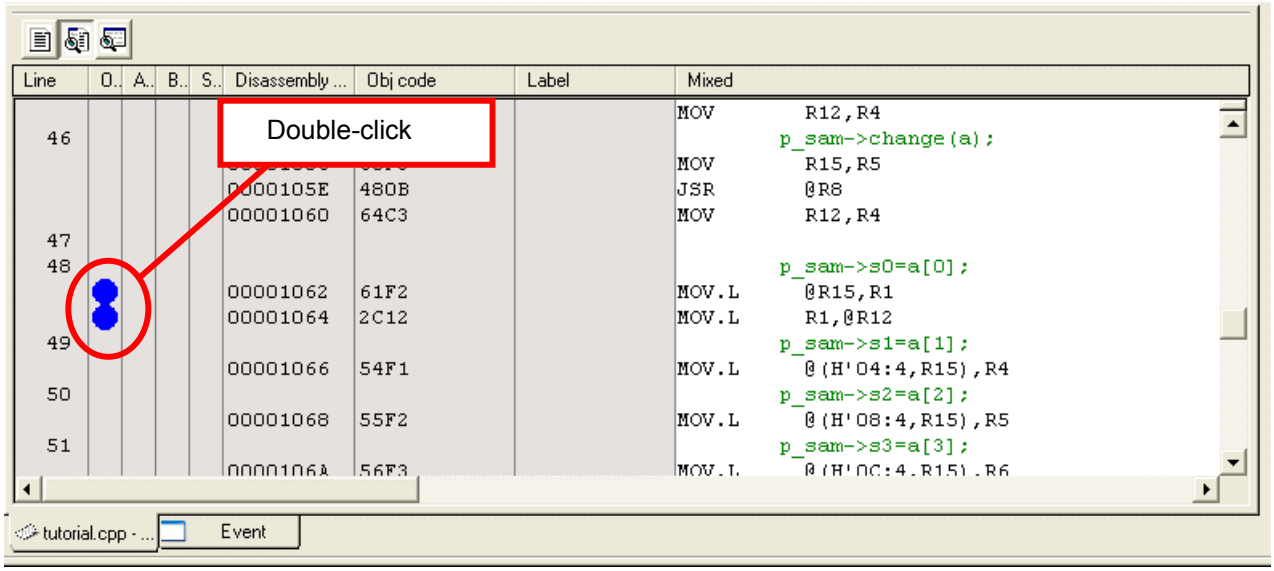


As is shown in the picture above, the performance analysis can also be done in the order [Ch 2 to Ch 1 PA].

4.5 Analyzing Performance by Measuring the Number of Executed Instructions

This section introduces how to specify [Number of execution instructions (I)] as the condition for a single assembly instruction. Here, this is used to measure the number of times a given instruction is passed.

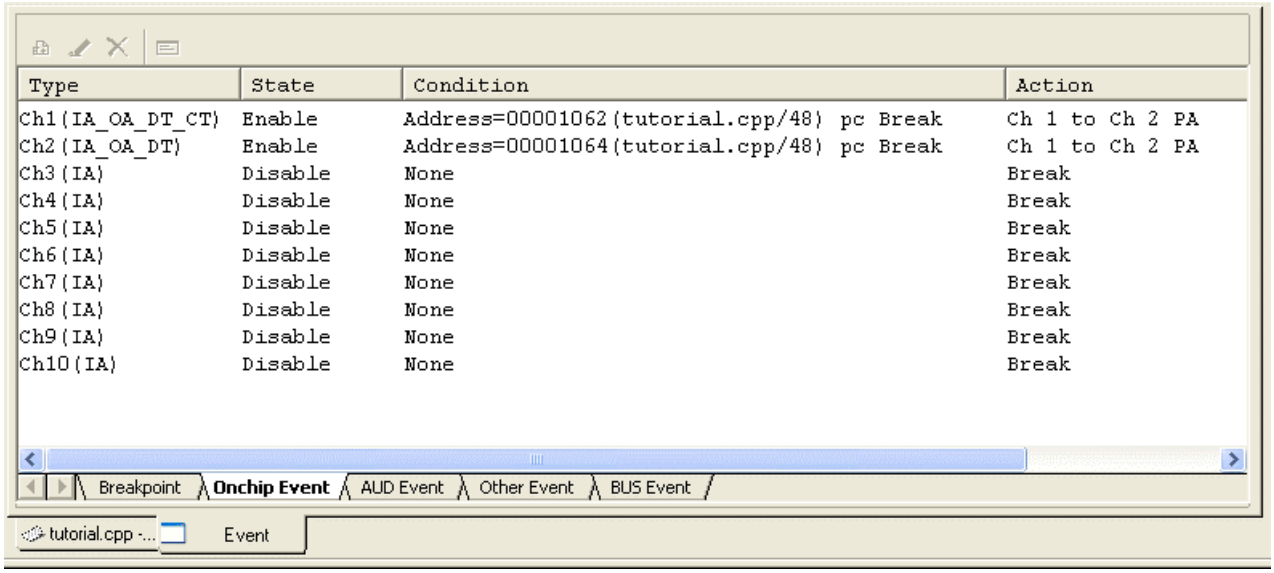
(1) View the source file “tutorial.cpp” in mixed mode and double-click on the [Onchip Event] column between lines 48 and 49 to set an on-chip breakpoint.



By specifying the first address of an instruction that is not a branch or delayed-slot instruction and the address of the following instruction as the condition, you will be able to view, in the [Result] column, the number of times this range has been passed. If you set a condition of this type in an interrupt handling routine, for example, it is possible to view the number of times a specific function is called.

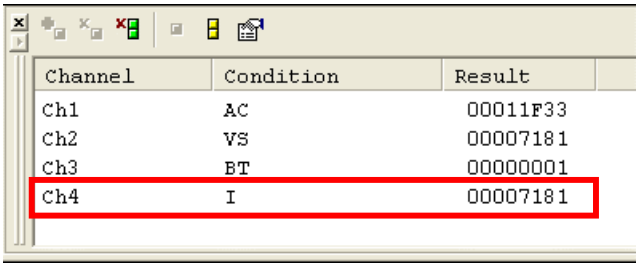
Note, however, that the number measured may not be correct if the target instruction is in an area where interrupts are enabled or the instruction may generate an exception.

(2) The [Event] pane shows the settings that have been made.



Specify the address range from H'00001062 to H'00001064 as the condition.

(3) After the program has been run, the [Performance Analysis] pane displays the results of analysis. The [Result] column for Ch4 ([I] condition) shows the number of times the instruction was executed.



In the screenshot above, the result is H'7181 (D'29057), which means address H'00001062 has been passed 29,057 times.

We conclude the description of performance analysis here.

5. Frequently Asked Questions

5.1 Can you provide any notes on analyzing performance?

(1) Use of the profiling and on-chip performance analysis functions at the same time is not possible. The [Can not use this function] error-message dialog box will be displayed if simultaneous use is attempted.

(2) Do not change the setting for performance analysis using channels 1 and 2 ([Ch 1 to Ch 2 PA] or [Ch 2 to Ch 1 PA] in the [Ch1,2,3] list of the [Combination action(Sequential or PtoP)] dialog box) on the [Onchip Event] tabbed page while the user program is running. Changing the settings in this way will lead to incorrect performance analysis.

(3) If the start condition is satisfied after the end condition has been satisfied, performance analysis will be resumed. The result shown at a break will be the sum of all results measured during the period of performance analysis.

(4) When you use the measurement start/end conditions ([Ch 1 to Ch 2 PA] or [Ch 2 to Ch 1 PA]), the value in the [Count] page of the [Event condition 1] dialog box must be specified as one.

(5) If there is a conflict between a DMA or DTC transfer and satisfaction of an event condition (including an external-bus access condition), this may prevent the triggering of a break, stopping or acquisition of internal trace information, or starting or stopping of performance analysis by the event condition.

(6) In debugging of the SH7125/SH7124 without the EV-chip unit connected to the emulator, the performance analysis function is not supported. That is, this function is only supported when the EV-chip unit is connected to the emulator.

6. Related Documents

The E200F emulator and High-performance Embedded Workshop provide many other useful functions not mentioned in this document. Please refer to the following related documents for important information such as detailed specifications, technical information, or restrictions on each product.

【Documents Related to the E200F Emulator】

- SH-2A, SH-2 E200F Emulator User's Manual
- Limitations on SH-2A, SH-2 E200F Emulator
- SH-2A, SH-2 E200F Emulator Additional Document for User's Manual
  (Supplementary Information on Using the SH7086)

【Documents Related to High-Performance Embedded Workshop】

- High-performance Embedded Workshop User's Manual
- High-performance Embedded Workshop Release Note

【Documents Related to MCU】

- SH7080 Group Hardware Manual
- SH-2A, SH2A-FPU Software Manual

【Document Related to SuperH™ Family C/C++ Compiler Package】

- SuperH™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package User's Manual
- SuperH™ RISC engine C/C++ Compiler Package Application Note

Visit the following Renesas websites for information on this product:

Global site: http://www.renesas.com/e200f

Japanese site: http://japan.renesas.com/e200f

Renesas Website and Customer Support

Renesas Technology Website:
http://www.renesas.com/

Customer Support:
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 12, 2008	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   1) artificial life support devices or systems
   2) surgical implantations
   3) healthcare intervention (e.g., excision, administration of medication, etc.)
   4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

© 2008. Renesas Technology Corp., All rights reserved.